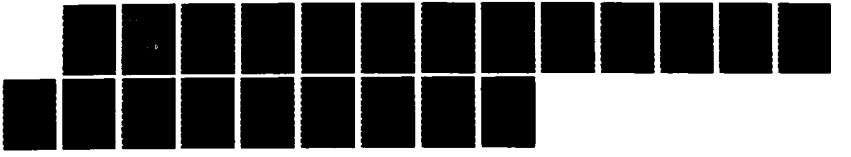
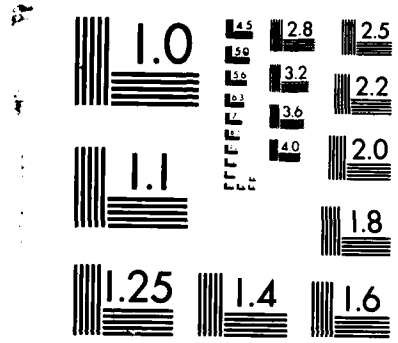
AD-A174 542 EVALUATION REPORT OF CODERCARD CPP-300 PORT PROTECTOR 1/1
(U) NATIONAL COMPUTER SECURITY CENTER FORT GEORGE G
MEADE MD M W HALE ET AL 07 APR 86 CSC-EPL-86/002
UNCLASSIFIED F/G 9/2 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

CSC-EPL-86/002

FINAL EVALUATION REPORT OF CODERCARD CPP-300 PORT PROTECTOR

7 April 1986

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Public Release; Distribution Unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CSC-EPL-86/002	5. MONITORING ORGANIZATION REPORT NUMBER(S) S228,219
6a. NAME OF PERFORMING ORGANIZATION National Computer Security Ctr.	6b. OFFICE SYMBOL (If applicable) C12
6c. ADDRESS (City, State and ZIP Code) 9800 Savage Road Ft. Meade, MD 20755-6000	7a. NAME OF MONITORING ORGANIZATION 7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable) 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.

11. TITLE (Include Security Classification) Final Evaluation Report, Codercard CPP-300 Port Protector	
12. PERSONAL AUTHOR(S) Hale, Michael; Behling, Barbara; Neufeld, Leon	
13a. TYPE OF REPORT Final	13b. TIME COVERED FROM _____ TO _____
14. DATE OF REPORT (Yr., Mo., Day) 86/04/07	15. PAGE COUNT 19
16. SUPPLEMENTARY NOTATION	

17. COSATI CODES		
FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Computer Security, Evaluated Products List (EPL), National Computer Security Center (NCSC), Trusted Path, User Identification and Authentication

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This is a report of the National Computer Security Center's (NCSC) evaluation of Codercard's CPP-300 product. The product performed a 2-way handshaking algorithm between the host site and the remote site and is designed to protect a single asynchronous communication prot from unauthorized use. While the CPP-300 does not encrypt the data that goes across the communication lines and does not implement a trusted path as described in the Department of Defense Trusted Computer System Evaluation Criteria, it can provide immediate improvements in security at computer installations which face attack from unauthorized users or computer "hackers".

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS ☐	21 ABSTRACT SECURITY CLASSIFICATION
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE NUMBER (Include Area Code) 22c OFFICE SYMBOL

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE

FOREWORD

This publication, CODERCARD CPP-300 Final Evaluation Report, is being issued by the National Computer Security Center under the authority and in accordance with DoD Directive 5215.1, "Computer Security Evaluation Center". The purpose of this report is to document the results of a component evaluation of the CODERCARD CPP-300 Trusted Path Port Protector.

FINAL REPORT
OF THE
CODERCARD CPP-300 TRUSTED PATH PORT PROTECTOR
COMPONENT EVALUATION

TABLE OF CONTENTS

EVALUATION TEAM MEMBERS

Barbara Behling
Michael W. Hale
Leon Neufeld

National Computer Security Center
9800 Savage Road
Fort George G. Meade, Maryland 20755-6000

EXECUTIVE SUMMARY

The National Computer Security Center has completed a component evaluation of the CODERCARD CPP-300 Trusted Path Port Protector - a computer system component that is intended to protect a single asynchronous communications path between computers, or equivalent devices, such as terminals. Since the CODERCARD CPP-300 is a component, and not a complete computer system, it was not evaluated against an entire class in the Department of Defense Trusted Computer System Evaluation Criteria, hereafter referred to as the Criteria. Rather, it was assessed as to how well it implements a trusted path and how well it performs user identification and authentication.

The NCSC evaluation team has determined that the CPP-300 does not implement a trusted path to a trusted computer system as described in the Criteria. The CPP-300 assures the user that he or she is communicating with the correct host computer's CPP-300 but it provides no assurance that the user is communicating with the trusted software on that computer.

On the other hand, the evaluation team feels that the CPP-300 is a valuable asset to an environment where identification of users is essential to accessing a computer system. The CPP-300 complements any identification and authentication mechanism that may already be present on the host computer system. It adds another barrier of protection to the computer system, supplementing password protection mechanisms. The NCSC does not recommend that the CPP-300 be used as a replacement for other identification and authentication mechanisms, but rather as an additional barrier to deny unauthorized users access to computer systems.

It should be noted that the CPP-300 does not encrypt data that goes over communication links. It generates pseudorandom numbers and performs a two-way handshaking protocol that helps to authenticate the identity of both ends of the communication link. Thus it does not provide any protection against eavesdropping. Codercard, does provide another product that encrypts all data transmitted, but that product has not been evaluated by the NCSC.

The CPP-300 can provide immediate improvements in security at computer installations which face attack from unauthorized users or computer "hackers".

INTRODUCTION

Background

On January 2, 1981, the Director of the National Security Agency was assigned the responsibility for increasing the use of trusted computer security products within the Department of Defense. As a result, the DoD Computer Security Center (CSC) was established at the National Security Agency. Its official charter is contained in DoD Directive 5215.1. In September 1984, National Security Decision Directive (NSDD 145) expanded these responsibilities to include all federal government agencies. At that time the Center became known as the National Computer Security Center (NCSC).

The primary goal of the NCSC is to encourage the widespread availability of trusted computer systems, that is, systems that employ sufficient hardware and software integrity measures for use in the simultaneous processing of a range of sensitive or classified information. Such encouragement is brought about by evaluating the technical protection capabilities of industry and government developed systems, advising system developers and managers of their systems' suitability for use in processing sensitive information, and assisting in the incorporation of computer security requirements in the systems acquisition process.

The NCSC Component Evaluation Program

While the NCSC devotes much of its resources to encouraging the production and use of large scale, multipurpose trusted computer systems, there is a recognized need for guidance on, and evaluation of, computer security products that do not meet all of the feature, architecture, or assurance requirements of any one security class or level of the Criteria. The NCSC has, therefore, established a Component Evaluation Program.

The goal of the NCSC's Component Evaluation Program is to provide computer installation managers with information on components that would be helpful in providing immediate computer security improvements in existing installations.

Components considered in the program are special purpose products that can be added to existing computer systems to increase some aspect of security and that, further, have the potential of meeting the needs of government departments and agencies. For the most part, the scope of a Component Evaluation is limited to consideration of the component itself, and does not address or attempt to rate the overall security of the processing environment or computer system that may be a potential application for the

component. To promote consistency in evaluations, and where appropriate, an attempt is made to assess a component's security-relevant performance in light of applicable standards and features outlined in the Criteria. Additionally, the evaluation team reviews the vendor's claims and documentation for obvious flaws which would violate the product's security features, and verifies, through functional testing, that the product performs as advertised. Upon completion, a summary of the evaluation report will be placed on the Evaluated Products List.

The report will not assign a specific rating to the product, but will provide an assessment of the product's effectiveness and usefulness in increasing computer security.

Testing Summary

Testing is a significant portion of a component evaluation. The software test script that was developed by the team attempted to test each major CODERCARD function. The CODERCARD test script was based on the Codercard Users Guide Trusted Path Port Protection Manual, August 1, 1985, which is a relatively high level description of the product. This testing emphasized quality assurance and not penetration of the component.

The results of the evaluation team's testing are fully reported in Appendix A of this report.

PRODUCT DESCRIPTION

CPP-300 Trusted Path Port Protector Overview

The Codercard Trusted Path Port Protection (CPP-300) device is designed to provide trusted path and identification and authentication protection over dial-up or dedicated lines. The configuration evaluated was specifically designed to fulfill the trusted path requirements of the DOD Criteria (CSC-STD-001-83).

CPP-300 Description

The Codercard CPP-300 trusted path port protector is designed to operate in pairs, and protect a single asynchronous communications path between computers or between computers and terminals. It provides stand alone protection, and is useable on either remote dial-in lines or dedicated lines. CPP-300 units are installed 'in-line', that is, between the protected terminal and the communication path at each end of the communication path.

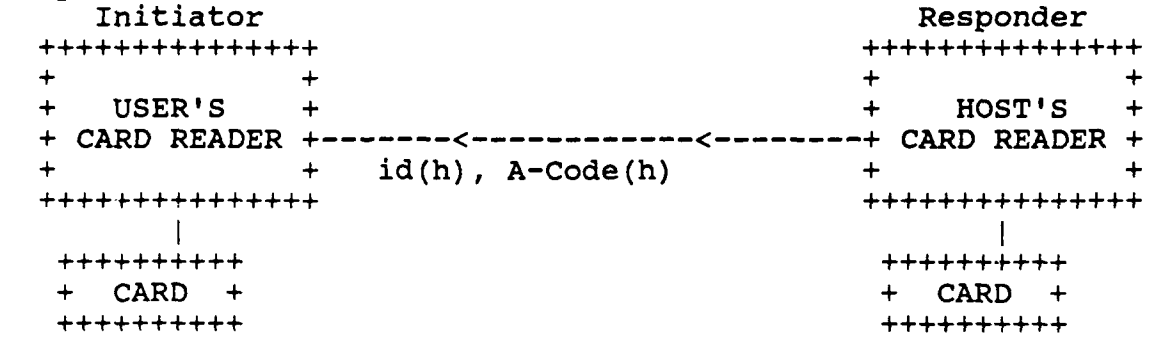
Each CPP-300 consists of an outboard Card Reader (CB-300) and a user Codercard (CC-100). Each Card Reader will remain in an unauthorized state and will not allow the start of the authentication procedure until the Codercards have been inserted in the Card Readers. Direct communication between the terminal and host is not permitted until identification and authentication is successfully completed. This is achieved through the execution of a proprietary non-linear random number generator.

In order for the authentication to be successful, the Codercard at one end of the communication line exchanges a series of 32 bit random numbers with the Card Reader at the other end. In this way, one Card Reader will authenticate the opposite Codercard. The random numbers generated are used to authenticate the Codercard at the terminal end with the Card Reader at the host end, and the Codercard at the host end with the Card Reader at the terminal end. Only after both exchanges have been completed successfully is access to the communication line granted.

Verification Protocol

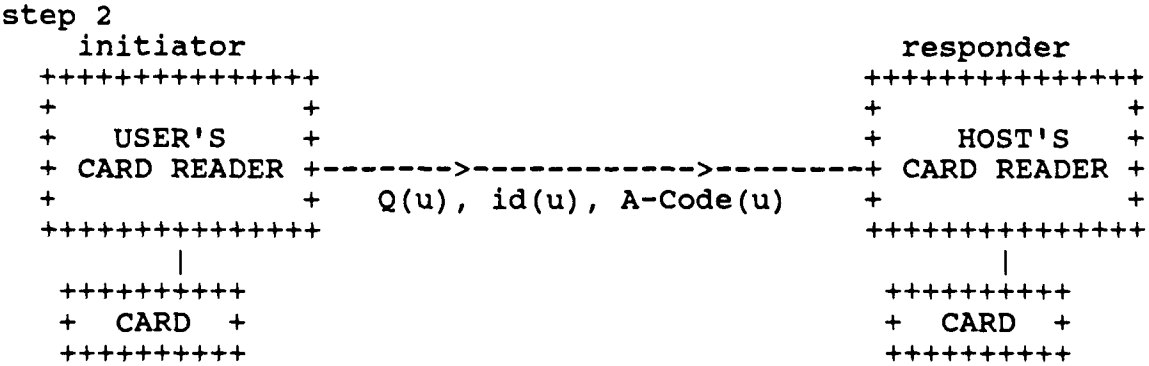
Following is a step-by-step explanation of how the verification handshake is performed. The protocol is performed when the Responder receives a request for authentication from the Initiator. While the following example depicts the remote user as the Initiator, it should be noted that either the Host or the User can initiate the authentication handshake and, hence, perform the role of the Initiator.

step 1



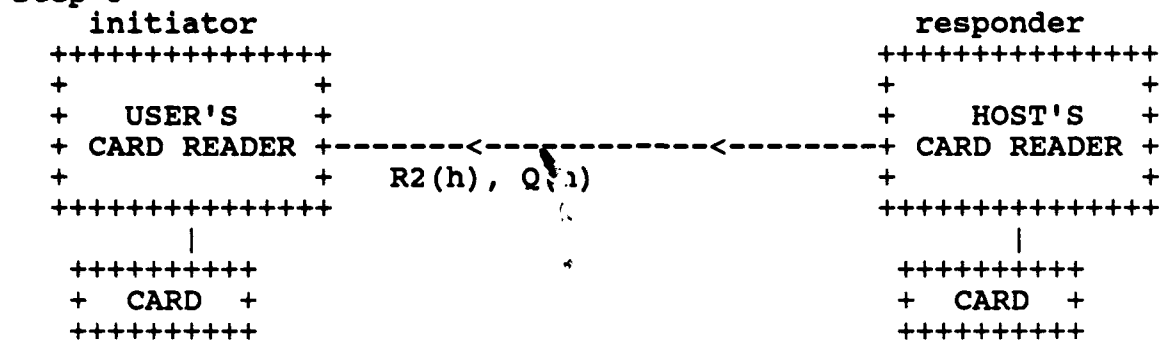
In step 1 the id and A-code of the host's Codercard is sent to the user end. The user's Card Reader sends these values to the codercard in its receptacle which then computes two numbers, Q(u) and R1(u).

step 2



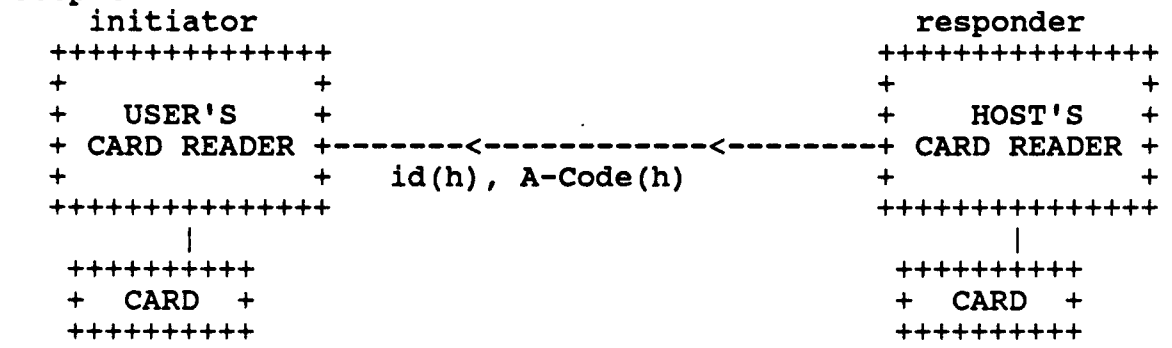
In step 2 the user's Card Reader queries its Codercard for its id(u) and A-Code(u) which it sends along with the Q(u) to the host's Card Reader. The user's Card Reader remembers R1(u) for later use.

step 3



In step 3 the host Card Reader passes Q(u) to the card in it's receptacle. The Codercard calculates Q(h) and R1(h) and R2(h). R2(h) and Q(h) are sent to the user. R1(h) is remembered for later use.

step 4



In step 4 the user Card Reader passes Q(h) to its Codercard. The Codercard calculates R2(u) and passes it back to the Card Reader at the host end. The host end now has R1(h) and R2(u). The user end has R1(u) and R2(h). For access to be granted, R1(h) must equal R2(u) and R1(u) must equal R2(h). Otherwise communication will not be established.

DOCUMENTATION OVERVIEW

This section contains a review of the documentation provided by Codercard for the Codercard CPP-300 Trusted Path Port Protector. Because there are no specific evaluation criteria for components, the team reviewed adequacy of documentation and tested the product to determine if it performed as stated in the documentation. The documentation that was reviewed for completeness and accuracy was entitled, "Users Guide Trusted Path Port Protector CB-300 & CB-305", dated 1 August 1985.

The Users Guide contains a description of the Codercard CPP-300 and sections detailing both host and remote commands for establishing and terminating access. In addition, sections that explain how diagnostics and the authentication protocol are performed are included. The documentation is technical in nature and is more suited for the system administrator than the occasional user. For example, the explanation of the Status Read command in section 4.7 diagrams the bit patterns for the status byte. The occasional user would benefit more from seeing a table of the codes "D0", "80", etc. with an English explanation than having to perform the specific bit translations. Also, the trusted path authentication protocol explanation in Appendix A is directed more toward a system administrator than the average user. The evaluation team understands that this is the only manual that applies to the product tested, but a separate manual that takes more of a cookbook approach would be appropriate.

The following paragraphs identify the places where discrepancies in the documentation were discovered:

(1) When describing the configuration of the switches in Section 6.0, the sixth bit was not explained. The documentation should state its function or that this bit is not used.

(2) The description of what happens when establishing access is inconsistent throughout the documentation. The description given in section 2.0 is inconsistent with the Command Summary Table of Section 4.8. Section 2.0 states that the unique id of the authenticated Codercard followed by an ASCII carriage return is displayed on the screen, while Section 4.8 states that the identification of the authenticated Codercard is displayed. The team found through testing that the status byte is what is actually returned.

(3) Upon execution of the terminate access <ESC><ESC>Ab command the team found that communication is disconnected in one direction only. While it is not possible to receive data at the end where this command was issued, it is still possible to send data for that end.

There are two places where the team found sections of the documentation to be particularly insufficient. The first was the explanation of the <ESC><ESC>A@ command. This command was poorly documented as the results obtained when the team tested the command were inconsistent with the documentation. Execution of this command caused a sixteen digit number to be transmitted back to and displayed on the terminal that initiated this command. The high order eight digits appeared to be the unique ID of the Codercard at the opposite end and the low order eight digits were interpreted by the team to be the A-code of the authentication protocol.

The team does not feel that the knowledge of the A-code is a significant vulnerability since one would have to tap the communication line to exploit this vulnerability. A line tap would expose all of the other numbers passed in the authentication protocol, making knowledge of the A-code insignificant.

The explanation of the trusted path authentication protocol in Appendix A is the other section that the team felt was poorly documented. When explaining how the codes get passed back and forth, it was not clearly defined whether the reference was to the responder or the initiator.

Although the documentation states that the Codercard Trusted Path Port Protector satisfies the trusted path requirement of the Department of Defense Trusted Computer System Evaluation Criteria (the Criteria), the CPP-300 was shown only to augment the Identification and Authentication requirement of the Criteria. A more detailed discussion of how the CPP-300 implements specific requirements in the Criteria can be found in the next section of this report.

USEFULNESS OF THE CPP-300

Trusted Path

The user's guide for the CPP-300 states that the device is designed to fulfill the Trusted Path requirement of the Department of Defense Trusted Computer System Evaluation Criteria. The evaluation revealed that the CPP-300 does not implement a trusted path as defined by the Criteria. The Criteria defines a trusted path as

> "A mechanism by which a person at a terminal can communicate directly with the Trusted Computing Base. This mechanism can only be activated by the person or the Trusted Computing Base and cannot be imitated by untrusted software."

In order to implement a complete trusted path, the mechanism must ensure that the user at the remote terminal is communicating with the trusted software on the host system. The assurance that an untrusted process is not masquerading as the trusted software (e.g. stealing user identification and authentication data) is imperative.

The CPP-300 does not provide the assurance that users are communicating with the trusted software on the host computer system. However, it does assure the user that he is communicating with the correct host system's CPP-300, and that the user accessing that system has a valid CODERCARD. These are desirable features in situations where computer systems are accessed via untrusted networks. Although this alone does not make the network trusted, it does provide some measure of assurance that users are communicating with the correct CPP-300 which should be connected to the correct computer system.

Identification and Authentication

Trusted computer installations must have the ability to uniquely identify each user who has access to the computer system. This is usually accomplished using public user identities and secret passwords to authenticate those identities. However, passwords can be compromised if they are not chosen properly or care is not taken to protect their secrecy.

The CPP-300 can be used as an additional barrier that users must penetrate before they are given access to the system. With the CPP-300, knowledge of another user's password is not sufficient to access the system. A subverter must also possess a valid CODERCARD to gain access to the system. The NCSC does not recommend, however, that the CPP-300 be used as the sole means of identification and authentication in any computer system. It should be used to augment the existing identification and authentication mechanisms on the host computer system.

The CPP-300 in a Trusted System Environment

The CPP-300 can be used as it comes from the manufacturer to enhance the security of any installation. It is especially useful when dial-up telephone lines are used to access the computer system. It will keep unauthorized users from getting past the communication ports with repeated penetration attacks. By stopping them at the communication ports, it prevents would-be penetrators from guessing authorized user's passwords. Additionally, it provides assurance over an untrusted network that users are communicating with the correct computer system's Codercard. Even with directly connected, physically protected terminals, it helps keep unauthorized personnel from within the organization (such as the custodial crew) from accessing the system.

The manufacturer has not incorporated a mechanism into the CPP-300 that will invalidate a lost or stolen Codercard, although the unique identification of each Codercard could be used to flag an invalid card on the host system.

It must be emphasized that the CPP-300 does not encrypt data that goes over communication links. It merely generates pseudo random numbers and performs a handshaking protocol that helps to authenticate the identity of both ends of the communication link. Thus it does not provide any protection against eavesdropping. However, Codercard does provide another product that encrypts all data that is transmitted, which was not evaluated by NCSC.

The National Computer Security Center makes no claim about the robustness of the handshaking algorithm used by the CPP-300. The vendor claims that the algorithm is secure and cannot be discovered. The NCSC does not analyze algorithms, therefore we cannot verify this claim.

APPENDIX A
CODERCARD Test Results

This set of test procedures was based on the User's Guide for the CPP-300 Trusted Path Port Protector provided to the evaluation team. Using the functionality specified for the CPP-300 in the User's Guide, the team developed the test plan and documented the results of each test.

The CPP-300 was tested in a simulated mode with two CPP-300 readers connected between two VT100 terminals. Since the CPP-300 system's basic purpose is to close or open the communication circuit between a terminal and host computer, the evaluation team viewed this as an acceptable test configuration.

NOTATION

The following notation is used in the remainder of the test results.

IC - Initial Condition -- the state the CPP-300 should be in before this test is executed.

TC - Test Case -- what is being tested

TA - Test Action -- what the tester should do

ER - Expected Results -- what should happen as described in the documentation

AR - Actual Results -- not always what should happen. The word, SAME, will be used to document actual results that are the same as the expected results.

CM - Comment

1.0 Establishing Access

This test group tests the different methods of establishing access through the CPP-300 to a computer system. The expected results were taken from section 2.0 of the documentation. Which differs from the Command Summary on page seven of the documentation.

1.1 TC: character typed from the terminal
IC: unauthorized, matching cards
TA: type character on terminal
ER: connection established
AR: SAME
CM: <escape> character will not work because CPP-300 is programmed to capture escape sequences (see test group 3).

1.2 TC: character typed from terminal
IC: unauthorized, non-matching cards
TA: type character on terminal
ER: connection not established
AR: SAME

1.3 TC: "<esc><esc>Aa" establishes access
IC: unauthorized, matching cards
TA: type "<esc><esc>Aa" on keyboard
ER: connection established, status byte returned
AR: connection established, "80" returned
CM: "80" is the status denoting a successful handshake.

1.4 TC: "<esc><esc>Aa" does not establish access without valid CODERCARD
IC: unauthorized, non-matching cards
TA: type "<esc><esc>Aa" on keyboard
ER: connection not established, bogus id of all ones returned
AR: connection not established, "D0" returned
CM: "D0" is the status for an unsuccessful handshake.

1.5 TC: "<esc><esc>Aa" maintains access if already authorized
IC: authorized, matching cards
TA: type "<esc><esc>Aa" on the keyboard
ER: connection maintained
AR: SAME

1.6 TC: CPP-300 goes idle after three unsuccessful access attempts
IC: unauthorized, non-matching cards, delay switch at 10 seconds
TA: type 3 characters on keyboard at 2 second intervals; put matching cards into CPP-300s and try to authorize
ER: will not be able to authorize for 10 seconds
AR: SAME

1.7 TC: CPP-300 goes idle for 2 minutes after three unsuccessful access attempts.
IC: unauthorized, non-matching cards, delay switch set at 2 minutes
TA: type 3 characters on keyboard at 2 second intervals; put matching cards into CPP-300s and try to authorize.
ER: will not be able to authorize for 2 minutes
AR: SAME

2.0 Terminating Access

2.1 TC: CPP-300 disconnects after no traffic timeout

2.1a IC: authorized, timeout set to 2 minutes
TA: Time period of no traffic until timeout
ER: disconnect after 2 minutes
AR: SAME

2.1b IC: authorized, timeout set to 8 minutes
TA: time period of no traffic until timeout
ER: disconnect after 8 minutes
AR: SAME

2.1c IC: authorized, timeout set to 16 minutes
TA: time period of no traffic until timeout
ER: disconnect after 16 minutes
AR: test not performed

2.2 TC: disconnect when carrier dropped
IC: authorized
TA: drop data carrier line
ER: disconnect
AR: test not performed

2.3 TC: "<esc><esc>Ab" command disconnects communication
IC: authorized
TA: type "<esc><esc>Ab" on keyboard
ER: communication disconnected
AR: Communication is disconnected but only in one direction. The end which issues the disconnect command can no longer receive data but can still send data.
CM: While this is not how it is explained in the documentation, it does not appear to affect security.

2.4 TC: "<esc><esc>Aa" command disconnects if non-matching card has replaced card used to authenticate.
IC: authorized, non-matching cards
TA: type "<esc><esc>Aa" on keyboard
ER: communication disconnected
AR: SAME

3.0 Host Control Commands

3.1 TC: authenticate remote user "<esc><esc>Aa"
IC: unauthorized, matching cards
TA: type "<esc><esc>Aa" on keyboard
ER: communication connection established
AR: SAME

3.2 TC: terminate access "<esc><esc>Ab"
IC: authorized
TA: type "<esc><esc>Ab" on keyboard
ER: communication connection broken
AR: see test case 2.3

3.3a TC: report id of remote CODERCARD "<esc><esc>Ac"
IC: authorized, matching cards
TA: type "<esc><esc>Ac" on keyboard
ER: id of remote CODERCARD returned
AR: SAME

3.3b TC: report id of remote card when card used to authorize has been replaced by an invalid card
IC: authorized, non-matching cards
TA: type "<esc><esc>Ac" on keyboard
ER: id of CODERCARD in remote CPP-300 returned
AR: id of CODERCARD used on original authentication returned.

3.4 TC: report hardware model and firmware version number "<esc><esc>Ad"
IC: authorized, matching cards
TA: type "<esc><esc>Ad" on keyboard
ER: hardware model and firmware version returned
AR: SAME

3.5 TC: reset "<esc><esc>Ae"
IC: authorized, matching cards
TA: type"<esc><esc>Ae" on keyboard
ER: unit at the end where command given is reset; communication disconnected
AR: unit is reset; communication is disconnected in one direction like the terminate access command (see 2.3)

3.6 TC: disable host commands "<esc><esc>Af"
IC: authorized, matching cards
TA: type "<esc><esc>Af" on keyboard; then try other host commands
ER: no further host commands are accepted
AR: SAME
CM: unit must reenter the unauthorized mode and be

authorized again before host commands will be accepted.

NOTE: These commands can be initiated from the terminal as well as the host.

4.0 Remote Commands

4.1 TC: trusted path initialization "<esc><esc>A@"
IC: authorized, matching cards
TA: type "<esc><esc>A@" on the keyboard
ER: no change
AR: returns 16 digit number the first eight of which are always the same while the second eight is always random-looking.
CM: This command is not well explained in the documentation and, hence, not well understood by the evaluation team.

4.2 TC: disable remote commands "<esc><esc>A?"
IC: authorized, matching cards
TA: type "<esc><esc>A?" on keyboard; then try other remote commands
ER: no further remote commands executed
AR: SAME

5.0 Internal Switch Settings

5.1 TC: switch setting to disable host commands
IC: N/A
TA: turn #2 on switch 2 off
ER: unit will not accept host commands
AR: SAME

5.2 TC: switch setting to disable remote commands
IC: N/A
TA: turn #3 on switch 2 off
ER: remote commands disabled
AR: SAME

☆ U.S. GOVERNMENT PRINTING OFFICE: 1986-622-823/408

END

12-86

DTIC